

# Prediction of Video Popularity in the Absence of Reliable Data from Video Hosting Services: Utility of Traces Left by Users on the Web

Alexey Drutsa  
Yandex\*  
adrutsa@yandex.ru

Gleb Gusev  
Yandex\*  
gleb57@yandex-team.ru

Pavel Serdyukov  
Yandex\*  
pavser@yandex-team.ru

November 28, 2016

## Abstract

With the growth of user-generated content, we observe the constant rise of the number of companies, such as search engines, content aggregators, etc., that operate with tremendous amounts of web content not being the services hosting it. Thus, aiming to locate the most important content and promote it to the users, they face the need of estimating the current and predicting the future content popularity.

In this paper, we approach the problem of video popularity prediction not from the side of a video hosting service, as done in all previous studies, but from the side of an operating company, which provides a popular video search service that aggregates content from different video hosting websites. We investigate video popularity prediction based on features from three primary sources available for a typical operating company: first, the content hosting provider may deliver its data via its API; second, the operating company makes use of its own search and browsing logs; third, the company crawls information about embeds of a video and links to a video page from publicly available resources on the Web. We show that video popularity prediction based on the embed and link data coupled with the internal search and browsing data significantly improves video popularity prediction based only on the data provided by the video hosting and can even adequately replace the API data in the cases when it is partly or completely unavailable.

**Keywords:** video; popularity prediction; embed; API data; crawled data; search logs; browsing logs; hosting provider; operating company

## 1 Introduction

With the stunning growth of user-generated content, we observe the constant rise of the number of companies that operate with web content not being services hosting it. In this respect, we can distinguish two types of companies. The first ones are the organizations that provide a hosting service for user content (*hosting providers*, or *HPs*). For instance, they are video hostings like Youtube, music sharing services like Soundcloud, etc. The second ones (*operating companies*, or *OCs*) are the organizations that operate with user content which is hosted externally at HPs or other OCs. Examples of operating companies are web search engine companies (e.g.,

---

\*16, Leo Tolstoy St., Moscow, Russia (www.yandex.com)

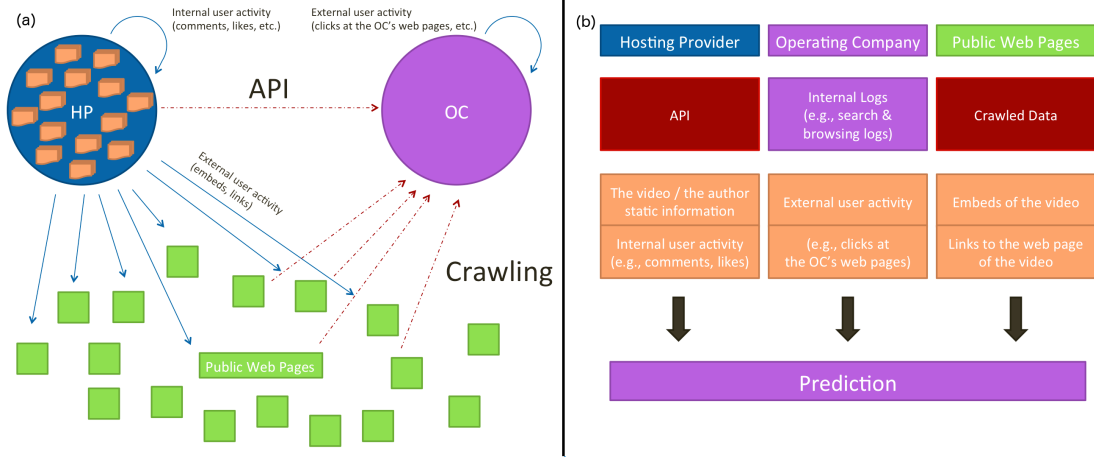


Figure 1: (a) A Hosting Provider (HP), an Operating Company (OC), and several public web pages that accommodate user activity about content items from the HP; (b) all available sources of evidence about current or future video popularity that could be split into the following three groups: API data provided by the HP, internal logs of the OP, and crawled data from publicly available web pages.

Google, Bing), content aggregators (e.g., Digg, Reddit), content recommendation systems (e.g., StumbleUpon, Pinterest), etc. Of course, one company may act both as HP and OC. For example, large social networks like Facebook and Twitter store billions of user messages and, at the same time, they provide the ability to embed external videos and images directly into the messages.

Since operating companies usually deal with tremendous amounts of external content, the challenge of estimating the current and the future popularity (e.g., the number of views, the number of comments received, etc.) of the content is inevitable for them. It is considered that the predicted current and future values of content popularity can serve as strong features for content ranking and content analysis problems in general [Gonçalves et al., 2010, Tatar et al., 2012, Yin et al., 2012, Ahmed et al., 2013]. So, a high quality popularity prediction is a vitally important component of any OC, which strongly influences the usefulness of the service to its end users and, consequently, the company’s profits.

In some situations, the popularity of the content is disclosed by the hosting provider through an application programming interface (API); in other circumstances it could not be retrieved from the HP at all (the API could be simply absent, as, for instance, for the video hosting services coub.com and break.com). Meanwhile, even if the API provides the information on popularity, the API could be periodically or permanently unavailable, or could set a limit on the number of allowed requests per time period, which can be insufficient for the OC’s needs. Besides, the provided data could be noisy and could be delivered with a delay as we demonstrate further with an example. In our work, we propose a methodology that could be used to compensate for the above limitations faced by operating companies.

In the current paper, we solve the popularity prediction problem and we restrict our investigation to the needs of a company which provides a popular video search service and aggregates content from different video hosting websites. Moreover, without a loss of generality, we will only consider the video data served by the video hosting website Youtube.

On the one hand, it is well known that the video popularity (the total number of views during a considered period) in vast majority of cases depends on or, at least, highly correlated with its popularity in the first days of its existence [Szabo and Huberman, 2010, Figueiredo et al., 2011, Pinto et al., 2013, Broxton et al., 2013]. Thus, the long-term popularity could be efficiently predicted using information about popularity dynamics in the first weeks of the video existence. On the other hand, it is known that Youtube can freeze the views count in the first days of video existence. It is officially confirmed by Youtube representatives<sup>1</sup> and is also frequently observed in the scope of the video search service of the popular search engine company under study. Thus, we face the problem of popularity prediction of a new video in the first days of its existence without knowing its previous popularity, because the information from Youtube is not always available.

In this paper, we propose to consider the case when an operating company decides not to stay completely dependent from video hosting providers and relies not on a single, but on all available sources of evidence about current or future video popularity that could be split into the following three groups (see Fig. 1):

- the data provided by the content hosting provider, generally, via its API or its publicly available web pages (API data);
- the internal data of the content operating company, generally, its user access data stored in logs (Log data);
- the publicly available resources of the Web, where the content could leave its traces (Web data).

The future popularity of objects of different kinds and videos in particular has been investigated and predicted from the point of view and using the data from content hosting providers only [Crane and Sornette, 2008, Szabo and Huberman, 2010, Lerman and Hogg, 2010, Tsagkias et al., 2010, Gonçalves et al., 2010, Lai and Wang, 2010, Hong et al., 2011, Borghol et al., 2012, Kupavskii et al., 2012, Kim et al., 2012, Radinsky et al., 2012, Pinto et al., 2013, Figueiredo, 2013, Ahmed et al., 2013, Broxton et al., 2013, Pinto et al., 2013, Ding et al., 2015, Fontanini et al., 2016, Li et al., 2016] and using internal data of social networks, such as Facebook and Twitter [Li et al., 2013, Soysa et al., 2013b, Soysa et al., 2013a, Abisheva et al., 2014].

A comprehensive overview of various research questions, methodologies and approaches in the field of prediction of web content (not only video) popularity can be found in [Tatar et al., 2014]. To the best of our knowledge, no existing study investigated the utility of publicly available traces of videos on web pages for the task of video popularity prediction. In the current paper, embeds of videos and links to video pages on publicly available resources of the Web are considered and used as features to predict video popularity. We also conduct a detailed investigation of the features extracted from all three groups of sources, where we use (the first to our knowledge) web search logs and browsing logs collected by Yandex ([www.yandex.com](http://www.yandex.com)) as internal data. The investigation of both sources and a series of experiments demonstrating to what extent both sources are capable to supplement or, more importantly, replace the API data in the task of video popularity prediction, *represent the first major contribution of this study*.

We are also the first who thoroughly investigated the prediction of popularity of a new video in the first days of its existence and the first who addressed the problem of the *current* video

<sup>1</sup>“We want to make sure that all views are validated so during this process the views update less frequently and might occasionally freeze above 300 views to assure quality view count. This is the normal operation in YouTube videos.” stated on the official Youtube site [support.google.com/youtube/troubleshooter/2991876](https://support.google.com/youtube/troubleshooter/2991876)

popularity prediction, in the absence of the information on popularity from the video hosting service. *We regard this as the second major contribution of this study.*

The rest of the paper is organized as follows. In Section 2, the related work is presented. In Section 3, we introduce our notations and the framework. In Section 4, prediction task is described in detail and the research questions are stated. We describe our data sets in Section 5, list the set of features in Section 6, and describe the models used for the popularity prediction in Section 7. In Sections 8 and 9, we present our experiments and discuss their results. In Section 10, the study’s conclusions and future work are presented.

## 2 Related work

We compare our research with other studies in three aspects. The first one relates to the video popularity analysis, the second one concerns the future video popularity prediction, and the last aspect refers to popularity prediction studies in general.

### 2.1 Video popularity analysis

The video hosting content and its popularity were widely investigated in recent years. In one of the most cited studies on the topic [Crane and Sornette, 2008], researchers focused on the analysis of video views dynamics decay after its peak. Both [Szabo and Huberman, 2010] and [Figueiredo et al., 2011] examined how quickly a video can become popular. They found that the most popular and viral videos receive the major part of their views in the first weeks of their existence. Another study [Broxton et al., 2013] also found that, in general, viral videos are mostly viewed in the first week after their upload. All the mentioned studies confirm the importance of studying and predicting video popularity in the first days of its existence.

### 2.2 Future video popularity prediction

The authors of [Szabo and Huberman, 2010] studied Youtube content popularity and established linear dependence between the logarithmic views counts measured at the 10-th day and at the 30-th day after the day of the video upload. The authors of [Ahmed et al., 2013] used the same data, but proposed to predict the future popularity by using a model of content propagation through an implicit graph induced by patterns of temporal evolution of video popularity. Prediction of the popularity peak day of a video was studied in [Jiang et al., 2014]. All described approaches are not applicable in our case, because, in order to predict future popularity, they exploit currently or/and previously observed popularity that is not always available by the statement of our problem.

The research described in [Li et al., 2013] was devoted to prediction of future video popularity in terms of the shares of a video in online social networks like Facebook. The analogous work was made in [Soysa et al., 2013b, Soysa et al., 2013a]: they collected the share data not being inside the social network company, but by receiving them from end users. Similar study of sharing behavior were carried out for Twitter [Abisheva et al., 2014]. Further studies concentrated on more sophisticated models and features like sentiments extracted from video frames and user comments [Ding et al., 2015, Fontanini et al., 2016]. The described methods could not be used in our work because they use either the data that is not publicly available for third parties, or the APIs of those social platforms. It means that a search engine which would decide to rely on that data would need to at least use the APIs of those services, while the goal of this study is to show to what extent an operating company can be independent from any APIs, even from seemingly more important APIs of video hosting services.

In Tables 1 and 2 we listed all features that we used to predict videos’ popularity. The contents of the table will be discussed in detail further in Section 6, but, at the moment, we are interested in the column named “Description (previously used or new)”, where we pointed out for each feature whether it has been used elsewhere in the literature, or it is a novel one. A reference is written in *italic style*, if the corresponding feature has been analyzed for a different purpose or has been used as a feature to predict the popularity of any object other than a video, but has *not* been used to predict videos’ popularity. Otherwise the reference is written in normal style.

The most comprehensive study of the features that could be retrieved via the Youtube API has been conducted in [Borghol et al., 2012] devoted to the content agnostic factors. Some of the features were also examined in [Figueiredo, 2013, Li et al., 2016]. Most of them are in Table 1, but not all: the researchers have used the number of keywords assigned to a video, the number of times the video was “favourited”, and the best quality the video is available in. All these features are still provided by Youtube API for old videos, but they seem to be deprecated for new videos: their values for all videos that were uploaded since 2013 are constant (zero, for instance). The reason is that Youtube does not allow its users to assign keywords and favorite videos anymore. In addition, the results of [Borghol et al., 2012] state that these features have no significant correlation with video popularity and the “favourited” feature has a strong correlation with other user feedback features (numbers of comments, ratings, likes). Therefore, we do not use them in our analysis. Nevertheless, these circumstances serve as an additional confirmation of any API inconsistency and indicate the high chance that the provided data could become obsolete or unavailable at any time.

Note that all above-mentioned studies and their corresponding features were used to predict future video popularity only, while our study also focuses on the prediction of current video popularity. Therefore, we reproduced all the features available through Youtube API and used them for our baseline models.

## 2.3 Popularity prediction of other objects

Prediction of web content (not only video) popularity is a well known and widely investigated problem [Tatar et al., 2014]. Prediction methods similar to the ones described in the previous subsection were applied to predict popularity of web pages in general [Szabo and Huberman, 2010, Lerman and Hogg, 2010, Hogg and Lerman, 2012] and for popularity of news measured in comments count [Tsagkias et al., 2010]. The popularity of tweets in terms of the number of retweets and shows were studied in [Hong et al., 2011, Kupavskii et al., 2012, Kupavskii et al., 2013].

Some studies consider popularity prediction models that utilize data in non-aggregated form (like news articles [Yang and Leskovec, 2010] and user comments [He et al., 2014]). The authors of [Yang and Leskovec, 2010] introduced the linear influence model in order to predict popularity of hashtags over Twitter network and popularity of memes over news articles and blog posts in terms of affected nodes of an implicit network. In our work, we use this model in order to take into account hosts with embeds and links to videos in non-aggregated form.

The usefulness of content popularity prediction for search engines was discussed in [Tatar et al., 2012, Yin et al., 2012, Tatar et al., 2014] where the prediction quality was estimated with ranking metrics for popularity of news articles and published jokes. Hence, in our study, we also evaluate the performance of our predictors by means of NDCG (one of the most popular ranking metrics [Järvelin and Kekäläinen, 2002]).

### 3 Notations and framework

Let  $[0, T)$ ,  $T > 0$ , be a known time interval, and  $\tau$  be a fixed time step (e.g., one day in our experiments). Then  $\tau$  induces the finite time grid  $\bar{\mathbb{T}} = \{t_m\}_{m=0}^M \subset [0, T)$ , where  $t_m = m\tau$ ,  $m = 0, \dots, M$ . The mesh without the starting point (e.g., the starting day) is denoted by  $\mathbb{T} = \bar{\mathbb{T}} \setminus \{0\}$ .

Let  $\mathcal{V}$  be a set of objects. Each object  $v \in \mathcal{V}$  is created at some time moment  $t_o(v)$  and exists at all times  $t \geq t_o(v)$ . From here on in the paper *we assume* that for each object  $v \in \mathcal{V}$ , there is chosen the object-specific time scale measured in days and centered in such a way that  $t_o(v) \in [0, \tau)$ , i.e. the object is created on the starting day of the scale. For instance, a video published at 18:00 7th May has  $t_o(v) = 0.75\tau$  for  $\tau = 1$  day. These object-focused timelines allow to consider examples of objects created at different days in the scope of the same prediction tasks we described further.

Each object can be represented by the features from some set  $\Phi$  during its life (for instance, video duration, number of comments, etc.). Each feature  $\varphi \in \Phi$  takes its values in a set  $\mathbb{D}_\varphi$ . Generally,  $\mathbb{D}_\varphi$  is the set of real numbers  $\mathbb{R}$  (e.g., for average rating), the set of integers  $\mathbb{Z}$  (e.g., for number of views), a finite set (e.g., for video category), their Cartesian product, or their subset. Besides, features could take different values at different time moments  $\bar{\mathbb{T}}$ , being dynamic in nature. Although, there are static features that are known either at or before the object creation time (such as video upload hour). Thus, each feature  $\varphi \in \Phi$  is formally a map from object set  $\mathcal{V}$  and the time mesh  $\bar{\mathbb{T}}$  to the feature value set  $\mathbb{D}_\varphi$ , that is  $\varphi : \mathcal{V} \times \bar{\mathbb{T}} \rightarrow \mathbb{D}_\varphi$ ,  $\varphi \in \Phi$ .

It is common that some data are known for the observer and some other are not. Usually, the observer wants to use some part of the known data to estimate or predict the unknown data. The set of features used to predict unknown data is denoted by  $\Psi \subset \Phi$ . The features that the observer wants to estimate or predict are referred to as *targets* and their set is denoted by  $\Theta \subset \Phi$ .

Let the values of features  $\Psi$  be known for the observer at the time moments  $\bar{\mathbb{T}} \cap [0, t_c]$ . Then the time moment  $t_c \in \bar{\mathbb{T}}$  is referred to as *the current time moment*. Consider that the observer has to estimate or predict the value of a target  $\theta \in \Theta$  at the time moment  $t_t \in \bar{\mathbb{T}}$ . Then the time moment  $t_t$  is called *the target time moment*. Note that, if  $t_c = t_t$ , then we are dealing with *current prediction*; if  $t_c < t_t$ , then we are dealing with *future prediction*<sup>2</sup>.

Thus, for a fixed target  $\theta \in \Theta$ , a fixed feature set  $\Psi' \subset \Psi$ , fixed current and target time moments  $t_c, t_t \in \bar{\mathbb{T}}$ , and a fixed prediction model  $\mathbf{m}$  there is a class of functions  $\mathfrak{P}_{\mathbf{m}}(\Psi', \theta)$  that predict the target  $\theta$  based on the features  $\Psi'$ . Then, the problem of prediction in terms of machine learning could be stated as follows. Given a training set of examples  $\mathcal{V}' \subset \mathcal{V}$  and a *prediction performance metric*  $\rho_\theta$  on the function space  $\{\mathcal{V}' \rightarrow \mathbb{D}_\theta\}$ , one should find the optimal predictor  $P_{\mathbf{m}, \text{opt}(\Psi', t_c; \theta, t_t)}$ , namely

$$P_{\mathbf{m}, \text{opt}(\Psi', t_c; \theta, t_t)} = \underset{P \in \mathfrak{P}_{\mathbf{m}}(\Psi', \theta)}{\text{argmin}} \rho_\theta \left( P|_{t=t_c}, \theta|_{t=t_t} \right). \quad (1)$$

### 4 Problem statement

In this section we present the research questions that we answer in our study, and we specify the prediction task by defining current and target days, target values that we aim to predict and the metrics that we optimize on the training data and measure on the test data.

<sup>2</sup>The current prediction task usually rises when the values of the target  $\theta$  could not be retrieved (e.g., the current number of views from Youtube API), while the future prediction task could be stated for any feature.

## 4.1 Research Questions

The main goal of our study is to identify the benefit of non-API data for the task of prediction of video popularity. Thereupon, we translate this objective into the following research questions:

- **[RQ1]** Could the prediction quality be improved by using the Web and the Log data in addition to the data from the video hosting?
- **[RQ2]** Could the Web and the Log data be effectively used in the case of absence of any reliable hosting data?
- **[RQ3]** Could the Web and the Log data replace a part of the hosting service data without a significant loss in prediction quality?

## 4.2 Specification of prediction task

In the paper, in accordance with the notations introduced in Section 3, our investigation object is a video uploaded to Youtube video hosting, and, thus, the set  $\mathcal{V}$  is a set of such videos. The fixed time interval  $\tau$  is equal to *one day*, and from here on in the paper we assume, for notation simplicity, that a unit of a time line is one day, i.e.,  $\tau = 1$ .

We investigate video popularity in terms of the number of views received by the video. This target could be defined in different ways, but we will consider the two most practical definitions of the target:

- *cumulative popularity* **Views[c]**: the total number of views received since the video creation, that is, in the time period  $[0, t)$ ;
- *daily popularity* **Views[d]**: the total number of views received during the last day, that is, in the time period  $[t - 1, t)$ .

In addition, both cumulative and daily views counts are logarithmically transformed<sup>3</sup> in order to better catch the differences between values of different magnitudes. Thus, the complete set of targets in our study is

$$\Theta = \{\mathbf{Views}[\mathbf{c}], \mathbf{Views}[\mathbf{d}], \log(\mathbf{Views}[\mathbf{c}]), \log(\mathbf{Views}[\mathbf{d}])\}. \quad (2)$$

In our work, we mainly focus on the task of the current popularity prediction ( $t_c = t_t$ ), given that the API does not provide us with this particular information (e.g., by delaying it) or given that the API is entirely or partly unavailable. However, we apply the framework to the prediction of the future popularity as well. For instance, for the 1-3 days forecast,  $t_t = t_c + \delta$ ,  $\delta \in \{1, 2, 3\}$ . We will consider target days of the two first weeks of video existence, that is,  $t_t \in \mathbb{T}^* \stackrel{\text{def}}{=} \{1, \dots, 14\}$ . The prediction in these target days is much more complicated than the prediction in the more distant time moments, because for large values of  $t_c, t_t$  (10 day  $\rightarrow$  30 day), the linear dependence of  $\log(\mathbf{Views}[\mathbf{c}])(t_t)$  and  $\log(\mathbf{Views}[\mathbf{c}])(t_c)$  was established [Szabo and Huberman, 2010, Pinto et al., 2013], which makes the prediction straightforward.

We use the *root mean squared error* (RMSE) as the minimization metric. RMSE for given maps  $P_1, P_2 : \mathcal{V}' \rightarrow \mathbb{D}_\theta$  is defined on a set  $\mathcal{V}' \subset \mathcal{V}$  by  $(\text{RMSE}(P_1, P_2))^2 = |\mathcal{V}'|^{-1} \sum_{v \in \mathcal{V}'} (P_1(v) - P_2(v))^2$ . The values of RMSE of the best predictor  $P_{\mathbf{m}, \text{opt}}(\Psi, t_c; \theta, t_t)$  on the test set are denoted by

$$\text{RMSE}(\Psi, t_c; \theta, t_t) = \text{RMSE}(P_{\mathbf{m}, \text{opt}}(\Psi, t_c; \theta, t_t), \theta). \quad (3)$$

<sup>3</sup>From here on in the paper we use  $\log(x) \stackrel{\text{def}}{=} \log_2(x + 1)$ , and, for each  $\varphi \in \Phi$ , we denote the logarithmic feature by  $\log(\varphi)$ .

We also analyze the performance of different predictors using a task-specific variant of ranking quality measure NDCG described in Section 8.3.

So, to finish the specification of prediction task one needs to specify the set of all investigated features  $\Psi$  and the used model  $\mathbf{m}$ . They will be specified in Sections 6 and 7 following the data sets description given in the next section.

## 5 Data sets

As we previously mentioned, we analyze three major data sources available for a typical OC. In our case, we used the following datasets:

- (a) content hosting provider (in our case, Youtube);
- (b) the internal logs of Yandex;
- (c) the traces that are left by the content on the Web.

The data from the hosting provider was split into two parts. The first part is the data about each video and its author provided through the Youtube API. The second part is the historical information on a video that is shown on the tab “Statistics” of the video web page (cumulative/daily views/shares/subscribers counts and total watch time)<sup>4</sup>. About 25% of videos are private or have no publicly available historical information. We collected only those videos for which the access was publicly available.

The data from Yandex could be also split into different parts corresponding to the logs of different services/products of the company. In our work, we investigate the logs of two services/products: the search service and the browser. The third major data source is provided by the web crawler of Yandex that supplies the two types of data about created videos: embeds of the videos and links to the videos’ pages.

So, our methodology for collecting data is as follows. First, we define a *harvest time period*. Then, at each day in this period we go to the database of the crawler (which also regularly crawls Youtube’s RSS feeds of the most popular and new videos<sup>5</sup>), and retrieve all available Youtube videos that are created in the defined time period. At the end of each day, we retrieve the data for these videos available via Youtube API. Thus, we obtain API information for each video for each day from the defined period.

When the harvest period is over, we retrieve the history of actual view counts per day for each known video. After that, we utilize search and browsing logs of Yandex and retrieve all available information about the known videos. Finally, we retrieve embeds of the videos and links to the videos from the crawler database.

**Dataset#1.** For the first dataset we selected the harvest time period from 23rd December, 2013 to 18th January, 2014 (27 days period). In that period we collected more than  $2.4 \cdot 10^6$  videos. The dataset has more than  $5.3 \cdot 10^6$  embeds of the videos (more than  $2.4 \cdot 10^5$  videos have at least one embed). The joint distribution of videos by the number of views and the number of embeds is presented in Fig. 2 (b).

**Dataset#2.** For the second dataset we selected the harvest time period from the 1st March, 2013 to the 31st May, 2013 (92 days period). Here we did not retrieve the API data for each day unlike for the previous dataset, because Dataset#2 is not used in the experiments with the API data. We collected it in order to train the linear influence model (see Section 7), and hence

<sup>4</sup>The data are provided with a delay of 2-3 days. The authors of [Figueiredo, 2013] and [Figueiredo et al., 2011] used the same methodology to extract views history.

<sup>5</sup>[http://www.youtube.com/t/rss\\_feeds](http://www.youtube.com/t/rss_feeds)



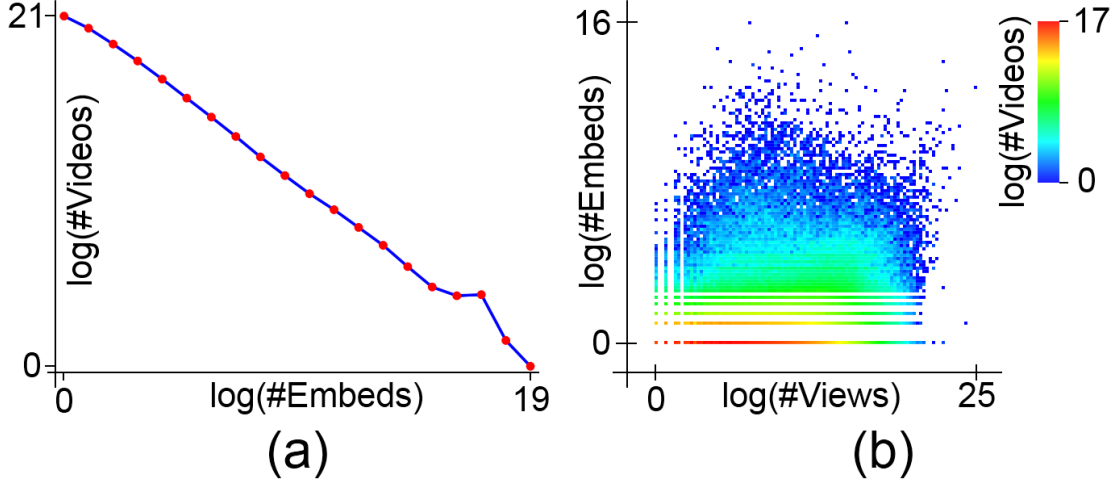


Figure 2: (a) The distribution of videos by the number of embeds obeys the power law (Dataset#2); (b) The joint distribution (heat map) of videos by the number of views (axis  $X$ ) and the number of embeds (axis  $Y$ ) for Dataset#1.

we also collected all embeds and links of the discovered videos that were published in the first 120 days after each video’s creation day. Overall, we collected more than  $8.5 \cdot 10^6$  videos. The data set has more than  $58 \cdot 10^6$  embeds of the videos (more than  $3.3 \cdot 10^6$  videos has at least one embed). The distribution of videos by the number of embeds obeys the power law, see Fig. 2 (a).

## 6 Features

We summarized and described all features used in our work in Tables 1 and 2. The sets  $\Psi$  that the feature belongs to are specified in the column “Feature set”. The feature value space  $\mathbb{D}_\psi$  is described in the column “Values”<sup>6</sup>. For each integer or real valued feature that has  $\gg 1$  values, we consider both its actual value and its *logarithmic* transformation (in terms of  $\log(x) \stackrel{\text{def}}{=} \log_2(x + 1)$ ). The logarithmic transformation of the feature  $\varphi \in \Phi$ ,  $\mathbb{D}_\varphi \subset \mathbb{R}$ , we denote by  $\log(\varphi)$ . For instance, **TitleLen** is the actual number of characters in the title of a video, and  $\log(\text{TitleLen})$  is the logarithm of that number. The presence of the mark “n/1” in Tables 1&2, column “Modes”, means that the marked feature is used in popularity prediction both in transformed and non-transformed forms. The column “Modes” of Tables 1&2 contains also marks “c/d”. The presence of the mark means that the corresponding feature possesses the integral property. It means that the feature value at a time moment  $t \in \mathbb{T}$  is equal to the sum of elementary features over some time period. In our work, we use two options: the value calculated during the entire period  $[0, t)$  (*cumulative*) and the value calculated during the last day  $[t - 1, t)$  (*daily*)<sup>7</sup>. In order to distinguish cumulative and daily feature values, we add to

<sup>6</sup>We remind the standard number set notations:  $\mathbb{N}$  is the set of natural numbers, i.e.  $\{1, 2, \dots\}$ ;  $\mathbb{Z}_+$  is the set of non-negative integer numbers, i.e.  $\{0, 1, 2, \dots\}$ ;  $\mathbb{R}_+$  is the set of non-negative real numbers, i.e.  $[0, +\infty)$ ,  $\mathbb{Z}_p, p \in \mathbb{N}$ , is the finite set of numbers  $\{0, 1, \dots, p - 1\}$ .

<sup>7</sup>It is possible to derive many other combinations like values calculated during the two previous days  $[t - 2, t)$  and so on. It is not the goal of our research, so we took just 2 different variants. A deeper investigation of other combinations could be regarded as future work.

Table 1: List of all features from API that are used to predict video popularity.

Feature set $\Psi$	Name $\psi$	Values $\mathbb{D}_\psi$	Description (previously used or new)	Modes
API <sub>S</sub>	<b>1. Static features from the video hosting service API</b>			
	<i>1.a. Static features about the video from video hosting service API</i>			
	API <sub>sv</sub>	Dur	$\mathbb{N}$ Video duration in seconds ([Figueiredo, 2013, Li et al., 2016])	n/1
		Cat	$(\mathbb{Z}_2)^c$ Video category, where $c \in \mathbb{N}$ is the number of different categories ([Figueiredo, 2013, Li et al., 2016], [Yang and Leskovec, 2010, Borghol et al., 2012, Pinto et al., 2013])	-
		TitleLen	$\mathbb{N}$ Video title length in number of characters ([Li et al., 2016])	n/1
		DescLen	$\mathbb{Z}_+$ Video description length in number of characters ([Li et al., 2016])	n/1
		UplDOW	$(\mathbb{Z}_2)^7$ Day of the week of the video upload date (new)	-
		UplHour	$[0, 24)$ The hour of the video upload time (new)	-
		<i>1.b. Static features from the video hosting service API about the video author</i>		
	API <sub>sa</sub>	AuthAge	$\mathbb{Z}_+$ The author’s age in number of days from her registration date ([Borghol et al., 2012, Li et al., 2016], [Kupavskii et al., 2013])	n/1
		AUplCnt	$\mathbb{N}$ The number of videos uploaded by the author ([Borghol et al., 2012], [Hong et al., 2011, Kupavskii et al., 2013])	n/1
		AViewSum	$\mathbb{Z}_+$ The total time in seconds that viewers watched all the author’s videos ([Borghol et al., 2012])	n/1
		FrndCnt	$\mathbb{Z}_+$ The number of the author’s friends ([Borghol et al., 2012, Li et al., 2016], [Kupavskii et al., 2013])	n/1
		SubsCnt	$\mathbb{Z}_+$ The number of the author’s subscribers ([Borghol et al., 2012, Li et al., 2016], [Kupavskii et al., 2013])	n/1
API <sub>D</sub>	<b>2. Dynamic features from the video hosting service API</b>			
		CommCnt	$\mathbb{Z}_+$ The number of all comments on the video ([Borghol et al., 2012], [Figueiredo, 2013])	n/1, c/d
		LikeCnt	$\mathbb{Z}_+$ The number of likes of the video ([Borghol et al., 2012])	n/1, c/d
		DislCnt	$\mathbb{Z}_+$ The number of dislikes of the video ([Borghol et al., 2012])	n/1, c/d
		MinRat	$\mathbb{Z}_5$ The minimum rating assigned to the video ([Borghol et al., 2012])	-
		MaxRat	$\mathbb{Z}_5$ The maximum rating assigned to the video ([Borghol et al., 2012])	-
		AvgRat	$[1, 5]$ The average rating assigned to the video ([Borghol et al., 2012])	-
		RatCnt	$\mathbb{Z}_+$ The number of ratings assigned to the video ([Borghol et al., 2012])	n/1, c/d
		Update	$\mathbb{Z}_+$ The number of days passed from the last update date ([Borghol et al., 2012])	n/1

Table 2: List of features from LOG and WEB that are used to predict video popularity (all of them are novel in the context of video popularity prediction).

Feature set $\Psi$		Name $\psi$	Values $\mathbb{D}_{\psi}$	Description (all of them are <b>new</b> )	Modes
LOG		<b>3. Dynamic features from logs of Yandex</b>			
	LOG <sub>S</sub>	<b>3.a. Dynamic features from search logs of Yandex</b>			
		ShowURL	$\mathbb{Z}_+$	The number of shows of the video URLs on SERP	n/l, c/d
		ClickURL	$\mathbb{Z}_+$	The number of clicks on the video URLs on SERP	n/l, c/d
		CTR	$[0, 1]$	The click-through rate of the video URLs on SERP	c/d
	LOG <sub>B</sub>	<b>3.b. Dynamic features from browsing logs of Yandex</b>			
BrowVisit		$\mathbb{Z}_+$	The number of visits of the video URLs registered in browsing logs	n/l, c/d	
WEB		<b>4. Dynamic features from the Web</b>			
	WEB <sub>ag</sub>	<b>4.a. Dynamic aggregated features from the Web</b>			
		EmbCnt	$\mathbb{Z}_+$	The number of all embeds of the video	n/l, c/d
		EmbHCnt	$\mathbb{Z}_+$	The number of all hosts with embeds of the video	n/l, c/d
		MaxEPerH	$\mathbb{Z}_+$	The maximum number of embeds of the video per host	n/l, c/d
		AvgEPerH	$\mathbb{R}_+$	The average number of embeds of the video per host	n/l, c/d
		MaxEPerP	$\mathbb{Z}_+$	The maximum number of embeds of videos per page	n/l, c/d
		AvgEPerP	$\mathbb{R}_+$	The average number of embeds of videos per page	n/l, c/d
		FirstEmb	$\mathbb{Z}_+$	The number of days passed since the first embed of the video	n/l
		LastEmb	$\mathbb{Z}_+$	The number of days passed since the last embed of the video	n/l
		AvgEmb	$\mathbb{R}_+$	The average number of days passed since any embed of the video	n/l
		LinkCnt	$\mathbb{Z}_+$	The number of all links to the video	n/l, c/d
		LinkHCnt	$\mathbb{Z}_+$	The number of all hosts with links to the video	n/l, c/d
		MaxLPerH	$\mathbb{Z}_+$	The maximum number of links to the video per host	n/l, c/d
		AvgLPerH	$\mathbb{R}_+$	The average number of links to the video per host	n/l, c/d
		FirstLink	$\mathbb{Z}_+$	The number of days passed since the day of the first link	n/l
		LastLink	$\mathbb{Z}_+$	The number of days passed since the video was linked last time	n/l
		AvgLink	$\mathbb{R}_+$	The average number of days passed since there was any link to the video	n/l
	WEB <sub>nag</sub>	<b>4.b. Dynamic non-aggregated features from the Web</b>			
		EmbedHost	$\mathbb{E}$	The host list with embed timestamps of the video (preprocessed into the outcomes of the LIM)	n/l, c/d
		LinkHost	$\mathbb{L}$	The host list with link timestamps of the video (preprocessed into the outcomes of the LIM)	n/l, c/d

the feature name suffixes “[c]” and “[d]”, correspondingly. For instance, `EmbCnt[c]` is the number of embeds that a video received during its existence up to the current time moment  $t_c$ , whereas `EmbCnt[d]` is the number of embeds that the video received during the last day before the current time moment  $t_c$ , that is  $[t_c - 1, t_c)$ .

We split the features from the hosting provider into the dynamic feature set  $\text{API}_D$  and the static feature set  $\text{API}_S$ . The static feature set consists of the set  $\text{API}_{sv}$  of features about the video and the set  $\text{API}_{sa}$  of features about the author of the video. The features from the sets `LOG` and `WEB` are dynamic. The feature set `LOG` extracted from the logs of Yandex, we split into the features from the search logs  $\text{LOG}_S$  and the features from the browsing logs  $\text{LOG}_B$ . The feature set `WEB` extracted from the publicly available web resources are based on monitoring both embeds and links to the videos. We split them into aggregated features  $\text{WEB}_{ag}$  and non-aggregated features  $\text{WEB}_{nag}$ .

Non-aggregated features are introduced only to use them within the linear influence model described in Section 7. The difference between aggregated and non-aggregated features consists in the following. An aggregated feature is a feature that aggregates the information about a number of elementary features, that are called *non-aggregated features*. For instance, the fact of the video’s embed(s) at a specific web site (host) can be represented by an elementary non-aggregated feature. Usually, because of their large number, such features are aggregated in a small number of features with each of them representing some aspect (e.g., the total number of hosts that have at least one embed of or a link to the video). In our paper, for a non-aggregated embed feature, we have feature value space  $\mathbb{E}$  that has the form  $\mathbb{E} = (2^{\mathbb{T} \times \mathbb{N}})^{c_e}$ , where  $c_e = |\mathcal{H}'|$  is the number of used hosts and each element  $(t, n) \in \mathbb{T} \times \mathbb{N}$  corresponds to the event that a particular host embeds a video  $n$  times at its  $n$  pages at the  $t$ -th day since the video creation. The feature value space  $\mathbb{L}$  in the case of links is of the same form as  $\mathbb{E}$ .

Finally, the table provides the information about which features were previously investigated in the literature devoted to video popularity (column “Description (previously used or new)”<sup>8</sup>). From that one can learn that feature sets  $\text{API}_{sa}$  and  $\text{API}_D$  are entirely previously investigated, while the feature set  $\text{API}_{sv}$  contains features that are not previously investigated. We unite previously investigated features of the set  $\text{API}_{sv}$  in the set denoted by  $\text{API}_{svb} = \{\text{Dur}, \text{Cat}, \text{TitleLen}, \text{DescLen}\}$ . Then all previously investigated features are united in the set denoted by  $\text{BASE.lit} = \text{API}_{svb} \cup \text{API}_{sa} \cup \text{API}_D$ . So, the set  $\text{BASE.lit}$  is our first baseline feature set (*related work based baseline*). At the same time, we will consider all features that we could extract from the data acquired from the hosting provider’s API (as of January 2014) as our second baseline feature set (*API baseline*), that is the set  $\text{API} = \text{API}_S \cup \text{API}_D$ . At last, we define a short synonymous notation for the set of all features  $\text{ALL} = \text{API} \cup \text{WEB} \cup \text{LOG}$ .

## 7 Models

### 7.1 General model

Since our main goal is to compare different features, we use the same prediction model for all features. We considered a state-of-the-art *Friedman’s gradient boosting decision tree model* [Friedman, 2001] and a traditional *linear regression model*. The model’s characteristics are described more precisely at the beginning of Section 8. For non-aggregated features we implement the **linear influence model**, which is described further in this subsection. The predictions of the linear influence model are combined with other aggregated features, that is, we use them as additional features of decision tree models.

<sup>8</sup>The *italic* style of the reference means that the feature was just investigated for some other task, but was not used to predict popularity.

## 7.2 Linear influence model

The *linear influence model* (LIM) was introduced in [Yang and Leskovec, 2010], where it was used to predict diffusion of hashtags over Twitter network and diffusion of memes (short textual phrases) over news articles and blog posts. The model’s implementation aspects and modifications were discussed later in [Wang et al., 2013]. The main advantage of the model consists in that it does not require any knowledge of the network structure where the information is spreading.

In our work, we consider a video  $v \in \mathcal{V}$  as some infection. It diffuses through implicit network of users and web sites (or *hosts*). A video infects users when they watch it and it infects hosts if they contain web pages with an embed of the video or a link to the video web page. So, the diffusion network could be represented as a bipartite graph, where the first layer of nodes is a set of users  $\mathcal{U}$  and the second layer of nodes is a set of web sites (hosts)  $\mathcal{H}$ .

Then, the linear influence model states that each node  $h \in \mathcal{H}$  possesses a particular influence function  $I_h : \{0, \dots, (L - 1)\} \rightarrow \mathbb{R}_+$ , where  $L$  is the size of the function *domain*. The value  $I_h(t)$  is equal to the number of nodes from  $\mathcal{U}$  that will be infected by the node  $h$  during the  $t$ -th day after the node  $h$  was infected, that is, during the time period  $[t_h + t - 1, t_h + t)$ , where  $t_h \in \mathbb{T}$  is the time moment when the node  $h$  was infected. Then the number of views of a video in a particular day equals to the outcomes of a sum of the influence functions of previously infected hosts (see [Yang and Leskovec, 2010] for details).

Thus, on the one hand, the number of views of a video  $v \in \mathcal{V}$  is exactly the number of times when the nodes from the set  $\mathcal{U}$  were infected. On the other hand, the operating company cannot observe particular user infections, but can observe a subset of web sites  $\mathcal{H}' \subset \mathcal{H}$  and fix the time when some of them embed the video (or create a link to the video web page), and hence become infected.

To the best of our knowledge, we are the first who introduced the application of the linear influence model to a bipartite graph, where each part has its own criteria for infection and where a prediction of infection spread of the one part is made based on the infection spread observed in the other. Since we investigated the case when negative values of influence functions are allowed (sometimes, the fact of an embed at a particular highly specialized host may indicate that the video will not be popular), hence the LIM prediction problem is reduced to the least squares problem with a sparse matrix. Thus, the LIM allows us to use video features in non-aggregated forms. The outcomes predicted by LIM are used as features of our general model (see Section 8).

## 8 Experimental setup

### 8.1 Model settings

As it was stated above, the main objective of our study is an investigation of the wide range of available data for the task of video popularity prediction. Thus, we use the same machine learning algorithm for all experiments conducted in our work, except for the case of non-aggregated features  $\text{WEB}_{\text{nag}}$ , where we use the LIM (see Section 7). In all described experiments of our work, we used a proprietary implementation of the gradient boosted decision tree-based machine learning algorithm [Friedman, 2001] with 1000 iterations and 1000 trees, which appeared to be the best settings on the validation data. During our experimentation, we also utilized the traditional *linear regression model*, however, this model demonstrated a considerably worse performance with respect to the decision trees (by a minimum margin of 10% in our experiments).

As it was stated in Section 7, the linear influence model has the following parameters: the size  $L$  of the influence function domain, the size  $|\mathbb{T}|$  of the learning time period, the learning set  $\mathcal{H}'$  of hosts, the training set  $\mathcal{V}'$  of videos (infections). Since prediction of video popularity with

Table 3: Baseline comparison in terms of the average normalized RMSE over first 14 days since video creation (in % with respect to API).

Features	Targets ( $\theta \in \Theta$ )			
	log-transformed		non-transformed	
	cumulative	daily	cumulative	daily
API	<b>0.356</b>	<b>0.435</b>	<b>0.822</b>	<b>0.91</b>
BASE.lit	+ <b>0.79%</b>	+3.02%	+ <b>0.81%</b>	+3.92%
API <sub>sv</sub>	+154.96%	+112.91%	+21.68%	+9.85%
API <sub>sa</sub>	+38.44%	+33.08%	+11.82%	+6.78%
API <sub>d</sub>	+53.54%	+26.99%	+2.26%	+1.73%
API <sub>sa</sub> $\cup$ API <sub>d</sub>	+4.03%	+2.95%	− <b>0.3%</b>	− <b>0.63%</b>
API <sub>sv</sub> $\cup$ API <sub>d</sub>	+33.75%	+16.45%	+2.67%	+1.62%
API <sub>sv</sub> $\cup$ API <sub>sa</sub>	+33.93%	+29.53%	+12.62%	+6.64%

LIM model is computationally expensive (though the problem matrix is sparse, what seriously decreases the number of elementary operations), we implemented the LIM solver on a distributed cluster system with the proprietary MapReduce technology [Dean and Ghemawat, 2008] that allowed to train the model using more than  $8 \cdot 10^6$  training videos in acceptable time.

We conducted a series of experiments in order to determine how the LIM parameters affect the prediction quality. We found that for different zones of the set of target days  $\mathbb{T}$  the model has different optimal parameters  $L$  and  $|\mathbb{T}|$ . Thus, we learn 12 LIMs on the Dataset#2 with different values of the parameters both for embed data and for link data. As a result, we obtained 6 influence functions per host (with the top-1280 hosts ranked by total number of embeds ( $|\mathcal{H}'| = 1280$ ), the domain size  $L \in \{1, 10, 20\}$ , and the sizes  $|\mathbb{T}|$  were chosen equal to optimal values per each  $L$ ). The outcome of each of these functions is a quadruple (non-/log-transformed cumulative/daily popularity). Thus, 12 LIM functions contributed 48 features studied in the experiments described further in Section 9.

## 8.2 Target and target days

In accordance with our problem statement (Section 4), we run our models for each of the target days  $t_t \in \mathbb{T}^*$ , i.e., for  $1, \dots, 14$  days since video creation time, and for all targets from  $\Theta$ , see Eq. (2). We address both types of prediction tasks:

- the current popularity prediction, i.e., where the current time moment equals to the target one:  $t_c = t_t$ ;
- the future popularity prediction with forecast days  $1, \dots, 13$ , i.e., where the current time moment is lower than the target one by an increment  $\delta$ :  $t_c = t_t - \delta$ ,  $\delta = 1, 2, \dots, t_t - 1$ .

## 8.3 Performance measures

Since, the values of RMSE are not normalized, they vary considerably depending on the target time  $t_t$ <sup>9</sup> and that could make it difficult to interpret results. Therefore, in the major part of the results we will normalize the RMSE values by the RMSE values of the baseline BASE.avg (see

<sup>9</sup>It is caused by a large difference in the number of views for different days. The variation of non-normalized RMSE could be seen in Figure 3 at the next experiment discussions.

Section 9.1 for its description) for each set of features  $\Psi$  and for each  $t_t \in \mathbb{T}^*$  and  $t_c = t_t - \delta$ ,  $\delta = 1, 2, \dots, t_t - 1$ :

$$\text{nRMSE}(\Psi, t_c; \theta, t_t) = \frac{\text{RMSE}(\Psi, t_c; \theta, t_t)}{\text{RMSE}(\text{BASE.avg}, t_c; \theta, t_t)}.$$

After normalization one could obtain the average normalized RMSE (AnRMSE) over all target days  $\mathbb{T}^*$ , e.g., for the current popularity prediction case  $t_c = t_t$ :

$$\text{AnRMSE}(\Psi; \theta) = |\mathbb{T}^*|^{-1} \cdot \sum_{t_t \in \mathbb{T}^*} \text{nRMSE}(\Psi, t_t; \theta, t_t).$$

As the second performance measure, we will use *normalized discounted cumulative gain* [Järvelin and Kekäläinen, 2002] (*NDCG@100*) over the top-100 results predicted as the most popular by our methods. We consider this measure as the most relevant to our study as it directly reflects the profit that an operating company (and, especially, a search engine) may receive from a high quality popularity prediction mechanism. In our work, the gain of a ranked video equals to  $1/(\text{pos} + 1)$  (where *pos* is the position of the video in the (ideal) list, in which videos are ranked by the actual target value) by the target value for the first 100 most popular videos, and equals to 0, if the ranked video does not belong the the top-100 videos of that ideal ranking.

## 8.4 Test and training data sets split

We split the Dataset#1 randomly into three equal parts as it is done in [Szabo and Huberman, 2010, Pinto et al., 2013, Ahmed et al., 2013] for video popularity prediction. The first part is used as the test data, the second one serves as the training data and the third part is used as the validation set. We repeated this procedure 20 times in order to apply the *paired two-sample t-test* and measure the significance level of the obtained results. All differences in the presented results have p-value  $< 0.05$ .

# 9 Experiment Results

Our results for forecasting (i.e.,  $t_c < t_t$ ) are very similar to the results for current popularity prediction (i.e.,  $t_c = t_t$ ). Hence, in the subsections where we compare feature sets and models (i.e., in Sections 9.1, 9.2, 9.3, and 9.5), we describe only the results for the latter one, which we consider a novel and more relevant task for our study. On the contrary, the analysis of the performance of future popularity prediction with different forecasting horizons and its comparison with the one of current popularity prediction are done in Section 9.4.

## 9.1 Baselines

Before the start of our investigation of the web and internal log data utility, we describe our baseline methods. We have implemented the following baselines:

- the predictions based on all API features (**API**);
- the predictions based on the API features used previously in the related studies (**BASE.lit**) (see Sections 2 and 6);

Table 4: Comparison of API, Web and Log feature sets in terms of the average normalized RMSE over first 14 days since video creation (in % with respect to API).

Features	Targets ( $\theta \in \Theta$ )			
	log-transformed		non-transformed	
	cumulative	daily	cumulative	daily
API	0.356(0%)	0.435(0%)	0.822(0%)	0.91(0%)
API $\cup$ LOG	−1.4%	−1.53%	−2.61%	−0.27%
API $\cup$ WEB	−1.19%	−0.8%	−10.15%	−4.36%
ALL	− <b>2.37%</b>	− <b>2.11%</b>	− <b>10.7%</b>	−4.28%
ALL $\setminus$ WEB <sub>nag</sub>	− <b>2.35%</b>	− <b>2.09%</b>	− <b>10.72%</b>	−4.75%
API $\cup$ WEB <sub>ag</sub>	−1.17%	−0.8%	−10.11%	− <b>5.29%</b>
LOG	+156%	+106.94%	+18.29%	+8.49%
WEB	+142.21%	+98.12%	+8.59%	+3.74%
WEB $\cup$ LOG	+132.74%	+89.07%	+4.64%	+3.75%
WEB <sub>ag</sub> $\cup$ LOG	+132.74%	+89.07%	+4.46%	+3.92%
WEB <sub>nag</sub>	+143.01%	+99.72%	+9.18%	+6.05%
WEB <sub>ag</sub>	+142.21%	+98.12%	+9.13%	+4.89%

- the naive average prediction model, which, for each video from the test data set, predicts the target value as the average of the corresponding target values on the training data set for all videos. This baseline is denoted by **BASE.avg**.

The average normalized RMSE values for the baseline methods are presented in Table 3 (except for **BASE.avg** whose AnRMSE is equal to 1). In the table, we compare the strength of feature sets **BASE.lit**, **API<sub>sv</sub>**, **API<sub>sa</sub>**, **API<sub>d</sub>**, **API<sub>sa</sub>  $\cup$  API<sub>d</sub>**, **API<sub>sv</sub>  $\cup$  API<sub>d</sub>**, and **API<sub>sv</sub>  $\cup$  API<sub>sa</sub>** by looking at the relative change of the metric against the full API feature set **API**. On the one hand, one could see that the set **API** outperforms the set **BASE.lit** for all targets and mainly in daily views, both transformed and not. On the other hand, the set **API<sub>sa</sub>  $\cup$  API<sub>d</sub>** makes the major contribution to the quality of prediction of the API set: it has no noticeable difference for non-transformed targets and loses a little bit on the **log**-targets. Thus, further in the paper we will use only the best baseline using all API features.

## 9.2 Web and logs vs API

The feature set **All** outperforms the API features. Thus, the web and log data notably improve the video popularity prediction quality in terms of all targets. This result could be seen in Table 4 and serves as the answer to the **RQ1**. In the same table, we compare all other feature sets with the API baseline and with each other.

One could see that the absence of API data has the most dramatic effect for *log*-transformed targets and only slightly reduces the video popularity prediction quality for non-transformed targets (**API** vs **WEB  $\cup$  LOG**). We conclude that the Web and the Log data could not completely replace API data for the purposes of the video popularity prediction in terms of exact values aggregated for 14 days. A completely different picture is observed for specific days (see further) and for ranking measures (see Section 9.5). At the same time one can see that the embed and link data (**WEB**) improve the quality better than the internal log data (**LOG**) of the operating company both independently (**WEB** vs **LOG**) and together with other features (**API** vs **API  $\cup$  WEB** vs **API  $\cup$  LOG**), (**ALL** vs **API  $\cup$  WEB** vs **API  $\cup$  LOG**). One could also see that the non-aggregated web features (**WEB<sub>nag</sub>**) used noticeably improve prediction quality of the aggregated (**WEB<sub>ag</sub>**) for



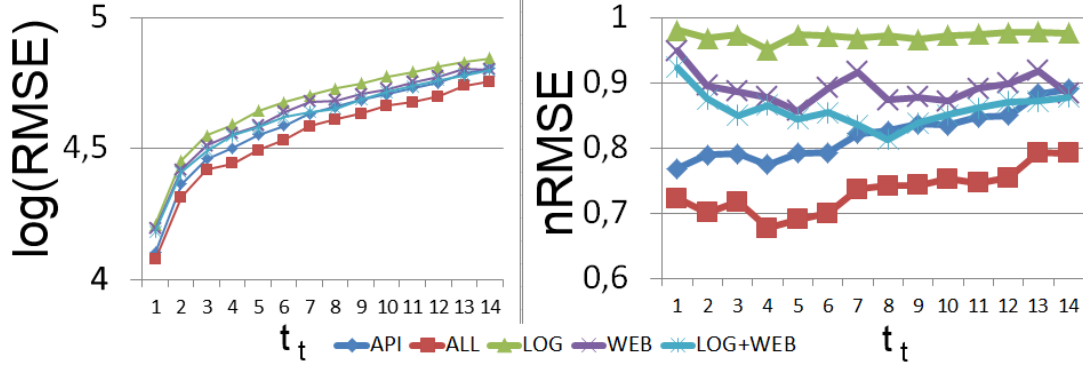


Figure 3: Comparison of API, Web and Log feature sets in terms of the RMSE (on the left) and the normalized RMSE (on the right) for each of the first 14 days since video creation for the target Views[c].

non-transformed targets: by 1.14% for daily and by 0.54% for cumulative views (WEB vs WEB<sub>ag</sub>). It is an important result given that we study the potential of an operating company to work in the absence of access to APIs.

In Figure 3, we demonstrate the values of the RMSE and the normalized RMSE measures per each day for the non-transformed cumulative target for the feature sets API, ALL, LOG, WEB, and WEB  $\cup$  LOG. Wherein, on the one hand, one could see that the API features lose quality with the growth of the number of the days passed since the video upload. The same situation is observed with the whole feature set ALL (which includes API features). On the other hand, the Web and Log data improve the quality of prediction of exact values of views with the growth of  $t_t$  at least to the end of the first week of the video existence, and *are able to compensate for the absence of API starting from the 8th day*. It is the partial answer to the **RQ2** (Section 9.5 has an extended and more definitive answer to **RQ2**).

### 9.3 Replacing parts of API with Web/Log data

We split the API feature set into the following groups (see the notations in Table 1):

- temporal context,  $\text{tc} = \{\text{Up1Hour}, \text{Update}, \text{Up1DOW}\}$ ;
- static video properties,  $\text{sv} = \{\text{Cat}, \text{Dur}, \text{TitleLen}, \text{DescLen}\}$ ;
- user feedback,  $\text{uf} = \{\text{Min/Max/AvgRat}, \text{Like/DislCnt}, \text{RatCnt}, \text{CommCnt}\}$ ;
- author rating,  $\text{ar} = \{\text{AuthAge}, \text{AUplCnt}, \text{AViewSum}\}$ ;
- social environment,  $\text{se} = \{\text{FrndCnt}, \text{SubsCnt}\}$ .

We measure the quality for the feature sets (API  $\setminus$  group) and (ALL  $\setminus$  group) by removing each group  $\in \{\text{tc}, \text{sv}, \text{uf}, \text{ar}, \text{se}\}$  from API and ALL feature sets respectively.

The obtained results for cumulative views<sup>10</sup> are presented in Table 5, where the columns “API  $\setminus$  group” show how much the prediction quality falls when a particular group becomes unavailable via API, and the columns “ALL  $\setminus$  group” show how much the prediction quality restores when we replace this group by the Web and the Log features. It is seen from the

<sup>10</sup>the results for daily views are similar

Table 5: Ablation of feature groups from API set with their further replacement by the Web and Log features. All results are in terms of the average normalized RMSE over first 14 days since video creation (in % with respect to the feature set API).

group	Cumulative number of views ( $\theta \in \Theta$ )			
	log-transformed		non-transformed	
	API \ group	ALL \ group	API \ group	ALL \ group
tc	+0.43%	−1.97%	+0.02%	−10.8%
sv	+3.95%	+1.28%	−0.8%	−10.81%
uf	+27.6%	+20.27%	+12.11%	−2.87%
ar	+18.6%	+14.59%	+1.14%	−9.91%
se	+3.73%	+0.82%	+1.56%	−9.53%

table that the absence of the *user feedback* feature group has the most dramatic effect for both targets. The absence of the *temporal context*, *static video properties*, and *social environment* feature groups has no significant consequences and could be easily replaced by the Web and the Log feature sets without any loss in the prediction quality. Finally, we conclude that *the Web and Log data could compensate for the absence of any of the studied parts of reliable API data without any loss in the quality* and even with a solid profit, in fact. It is the answer to the **RQ3**.

#### 9.4 Delay in data crawling

The prediction quality is also affected by the delay in data crawling. In order to study this influence, we fix the target day  $t_t$  and measure the quality for the feature set ALL, considering each day from the first day since video creation to this target day  $t_t$  as the current day  $t_c$ , i.e.,  $t_c = t_t - \delta$ ,  $\delta = 0, \dots, 13$ . In other words, we predict the current video popularity at the day  $t_t$ , as if we are in the past with data collected at the day  $t_c$ . The relative values of nRMSE (in terms of % with respect to the one for  $t_c = 1$ ) are presented in Fig. 4 for  $t_t = 7$  and 14, for each target  $\theta \in \Theta$ .

We see that the shorter the crawling delay  $\delta$  the better the prediction performance. However, for the target day  $t_t = 7$ , even a delay in one day leads to a noticeable quality drop (up to 2.5% for daily views), while, for the target day  $t_t = 14$ , such delay leads to a less dramatic quality drop (up to 1% for daily views). We conclude that *the speed of crawling (both of API, and of WEB data) has a strong influence on the popularity prediction performance, and the closer the prediction target day to the video creation moment, the more critical this influence is*.

#### 9.5 Ranking metric quality

One of the main applications of video popularity prediction is the proper ranking of the videos by their popularity. For instance, it allows the operating company to show the most popular videos on the main page, which always attracts a large share of user traffic<sup>11</sup>. Thus, we investigate how the quality measured in ranking metrics changes over predictions based on different feature sets. In Table 6, we present the average NDCG@100 over first 14 days for 7 main feature sets: API, ALL \ WEB<sub>nag</sub>, ALL, LOG, WEB<sub>ag</sub>, WEB, and WEB \ LOG.

As one could see, the feature set ALL clearly outperforms the feature set API for all listed targets, and especially when we optimize prediction of non-transformed targets. Moreover, one could see that WEB and WEB \ LOG notably outperform the baseline for cumulative targets, and

<sup>11</sup>e.g., Bing shows “most watched” videos at its main Video search page: <http://www.bing.com/videos/browse>

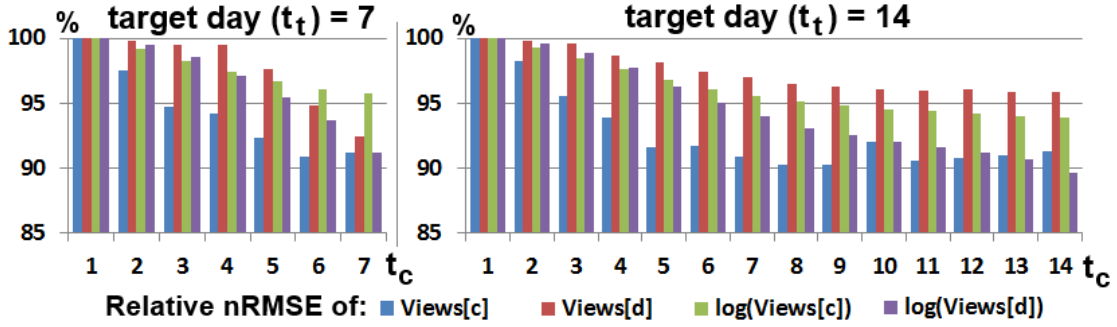


Figure 4: Prediction quality change of the feature set ALL with respect to different values of the current day  $t_c$ : from the first day since video creation to the target day  $t_t$  (7 or 14) (in % with respect to nRMSE of the data observed in the first day).

Table 6: Comparison of API, Web and Log feature sets in terms of average NDCG@100 over first 14 days since video creation for targets optimized by minimizing RMSE (in % of NDCG@100 with respect to API performance).

Features	Cumulative targets ( $\theta \in \Theta$ )				Daily targets ( $\theta \in \Theta$ )			
	log-transformed		non-transformed		log-transformed		non-transformed	
API	0.391	0%	0.334	0%	0.37	0%	0.361	0%
ALL \ WEB <sub>nag</sub>	0.466	+19.25%	0.506	+51.62%	0.398	+7.45%	0.495	+37.13%
ALL	0.426	+8.86%	<u>0.511</u>	+53.06%	0.398	+7.5%	<u>0.499</u>	+38.2%
LOG	<u>0.235</u>	-39.86%	0.136	-59.24%	<u>0.236</u>	-36.16%	0.137	-62.01%
WEB <sub>ag</sub>	0.435	+11.20%	0.406	+21.59%	0.337	-8.89%	0.332	-7.97%
WEB	0.442	+12.85%	0.405	+21.28%	<u>0.34</u>	-8.16%	0.337	-6.75%
WEB $\cup$ LOG	<u>0.464</u>	+18.48%	0.455	+36.4%	0.37	0%	<u>0.378</u>	+4.74%

WEB  $\cup$  LOG slightly outperforms the baseline or at least has the same value as the baseline for daily targets. *Thus, one could conclude that for purposes of ranking tasks the Web and Log data could completely replace the API data* (if a company does not have its own logs, then still Web data outperforms API data for cumulative views). It is the answer to the **RQ1** and **RQ2**. That means that if the task of an operating company is to present the lists of the most popular videos for the OC’s users, then: (a) it can be done independently from video service hostings, and (b) its costs on crawling and processing of embed and link data are justified.

Table 6 shows that implementation of the non-aggregated web features has a solid profit: their usage improves the prediction quality of aggregated features with respect to both the Web features only (WEB vs WEB<sub>ag</sub>: by 1.65% for cumulative, 0.73% for daily log-transformed views, and 1.22% for daily views without transformation) and all features (ALL vs ALL \ WEB<sub>nag</sub>: by 1.44% for cumulative and 1.07% for daily views without transformation).

From Table 6 one could also learn another lesson. Since the logarithmic transformation is monotonic, the original ranks of videos both in terms of the *log*-transformed and non-transformed target values are the same. Thus, the difference in average NDCG@100 values for the *log*-transformed and non-transformed targets gives the answer to the question: Does the log-transformation of views in the optimization of RMSE improve the ranking results? We underlined the best result between them for each feature set, independently both for cumulative and for daily views. One can conclude that *the transformation gives significant improvement for*

*some, but not all sets of features.* The improvement is more visible for cumulative views than for the daily ones.

## 10 Conclusions

We investigated the utility of newly proposed data sources (the embeds of and the links to videos publicly available on the Web and internal logs of Yandex) for the task of video popularity prediction and compared them with the data provided by the video hosting via its API. We prepared more than 100 features collected from all available data sources and to answer to a number of related research questions. We used both simple feature models, and more complicated feature models (the linear influence model that utilizes the features in non-aggregated form).

We found that the new data sources allow to improve the video popularity prediction quality, and they are able to compensate for the absence of API starting from the 8th day since the day of the video upload, if a company is interested in prediction of exact values of the current popularity. The new data could also compensate for any of the missing groups of API features that we considered in our study.

We also examined the case with a popular search engine, which predicts popularity of videos in order to present them in a proper order to its users. In that case we compared the relative performance between feature sets in terms of the ranking measure NDCG@100. We found that the web and log data could replace and even outperform the API data.

As future work we can, first, extend the set of feature source groups by investigating the data obtained from the API of another content operating company. Second, we can train different predictors for different topical categories of videos. Third, we can also experiment with training not the regression function and optimizing for RMSE, but a classifier to predict if the video belongs to the set of the most or the least popular ones. Finally, it would be interesting to study how to use online user feedback received on a list of the most popular videos presented on the main page in order to correct the prediction in real-time.

## References

- [Abisheva et al., 2014] Abisheva, A., Garimella, V. R. K., Garcia, D., and Weber, I. (2014). Who watches (and shares) what on youtube? and when?: using twitter to understand youtube viewership. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 593–602. ACM.
- [Ahmed et al., 2013] Ahmed, M., Spagna, S., Huici, F., and Niccolini, S. (2013). A peek into the future: predicting the evolution of popularity in user generated content. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 607–616. ACM.
- [Borghol et al., 2012] Borghol, Y., Ardon, S., Carlsson, N., Eager, D., and Mahanti, A. (2012). The untold story of the clones: content-agnostic factors that impact youtube video popularity. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1186–1194. ACM.
- [Broxton et al., 2013] Broxton, T., Interian, Y., Vaver, J., and Wattenhofer, M. (2013). Catching a viral video. *Journal of Intelligent Information Systems*, 40(2):241–259.

- [Crane and Sornette, 2008] Crane, R. and Sornette, D. (2008). Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences*, 105(41):15649–15653.
- [Dean and Ghemawat, 2008] Dean, J. and Ghemawat, S. (2008). Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.
- [Ding et al., 2015] Ding, W., Shang, Y., Guo, L., Hu, X., Yan, R., and He, T. (2015). Video popularity prediction by sentiment propagation via implicit network. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1621–1630. ACM.
- [Figueiredo, 2013] Figueiredo, F. (2013). On the prediction of popularity of trends and hits for user generated videos. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 741–746. ACM.
- [Figueiredo et al., 2011] Figueiredo, F., Benevenuto, F., and Almeida, J. M. (2011). The tube over time: characterizing popularity growth of youtube videos. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 745–754. ACM.
- [Fontanini et al., 2016] Fontanini, G., Bertini, M., and Del Bimbo, A. (2016). Web video popularity prediction using sentiment and content visual features. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 289–292. ACM.
- [Friedman, 2001] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.
- [Gonçalves et al., 2010] Gonçalves, M. A., Almeida, J. M., dos Santos, L. G., Laender, A. H., and Almeida, V. (2010). On popularity in the blogosphere. *IEEE Internet Computing*, 14(3):42–49.
- [He et al., 2014] He, X., Gao, M., Kan, M.-Y., Liu, Y., and Sugiyama, K. (2014). Predicting the popularity of web 2.0 items based on user comments. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 233–242. ACM.
- [Hogg and Lerman, 2012] Hogg, T. and Lerman, K. (2012). Social dynamics of digg. *EPJ Data Science*, 1(1):1–26.
- [Hong et al., 2011] Hong, L., Dan, O., and Davison, B. D. (2011). Predicting popular messages in twitter. In *Proceedings of the 20th international conference companion on World wide web*, pages 57–58. ACM.
- [Järvelin and Kekäläinen, 2002] Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.
- [Jiang et al., 2014] Jiang, L., Miao, Y., Yang, Y., Lan, Z., and Hauptmann, A. G. (2014). Viral video style: a closer look at viral videos on youtube. In *Proceedings of International Conference on Multimedia Retrieval*, page 193. ACM.
- [Kim et al., 2012] Kim, G., Fei-Fei, L., and Xing, E. P. (2012). Web image prediction using multivariate point processes. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1068–1076. ACM.

- [Kupavskii et al., 2012] Kupavskii, A., Ostroumova, L., Umnov, A., Usachev, S., Serdyukov, P., Gusev, G., and Kustarev, A. (2012). Prediction of retweet cascade size over time. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2335–2338. ACM.
- [Kupavskii et al., 2013] Kupavskii, A., Umnov, A., Gusev, G., and Serdyukov, P. (2013). Predicting the audience size of a tweet. In *ICWSM*.
- [Lai and Wang, 2010] Lai, K. and Wang, D. (2010). Towards understanding the external links of video sharing sites: measurement and analysis. In *Proceedings of the 20th international workshop on Network and operating systems support for digital audio and video*, pages 69–74. ACM.
- [Lerman and Hogg, 2010] Lerman, K. and Hogg, T. (2010). Using a model of social dynamics to predict popularity of news. In *Proceedings of the 19th international conference on World wide web*, pages 621–630. ACM.
- [Li et al., 2016] Li, C., Liu, J., and Ouyang, S. (2016). Characterizing and predicting the popularity of online videos. *IEEE Access*, 4:1630–1641.
- [Li et al., 2013] Li, H., Ma, X., Wang, F., Liu, J., and Xu, K. (2013). On popularity prediction of videos shared in online social networks. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 169–178. ACM.
- [Pinto et al., 2013] Pinto, H., Almeida, J. M., and Gonçalves, M. A. (2013). Using early view patterns to predict the popularity of youtube videos. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 365–374. ACM.
- [Radinsky et al., 2012] Radinsky, K., Svore, K., Dumais, S., Teevan, J., Bocharov, A., and Horvitz, E. (2012). Modeling and predicting behavioral dynamics on the web. In *Proceedings of the 21st international conference on World Wide Web*, pages 599–608. ACM.
- [Soysa et al., 2013a] Soysa, D. A., Au, O. C., Sun, L., Xu, L., Li, J., and Chen, D. G. (2013a). Advanced independent cascade model for youtube content propagation in facebook. In *Signal and Information Processing (ChinaSIP), 2013 IEEE China Summit & International Conference on*, pages 481–485. IEEE.
- [Soysa et al., 2013b] Soysa, D. A., Chen, D. G., Au, O. C., and Bermak, A. (2013b). Predicting youtube content popularity via facebook data: A network spread model for optimizing multimedia delivery. In *Computational Intelligence and Data Mining (CIDM), 2013 IEEE Symposium on*, pages 214–221. IEEE.
- [Szabo and Huberman, 2010] Szabo, G. and Huberman, B. A. (2010). Predicting the popularity of online content. *Communications of the ACM*, 53(8):80–88.
- [Tatar et al., 2012] Tatar, A., Antoniadis, P., De Amorim, M. D., and Fdida, S. (2012). Ranking news articles based on popularity prediction. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pages 106–110. IEEE Computer Society.
- [Tatar et al., 2014] Tatar, A., Antoniadis, P., De Amorim, M. D., and Fdida, S. (2014). From popularity prediction to ranking online news. *Social Network Analysis and Mining*, 4(1):1–12.

- [Tsagkias et al., 2010] Tsagkias, M., Weerkamp, W., and De Rijke, M. (2010). News comments: Exploring, modeling, and online prediction. In *European Conference on Information Retrieval*, pages 191–203. Springer.
- [Wang et al., 2013] Wang, Y., Xiang, G., and Chang, S.-K. (2013). Sparse multi-task learning for detecting influential nodes in an implicit diffusion network. In *AAAI*.
- [Yang and Leskovec, 2010] Yang, J. and Leskovec, J. (2010). Modeling information diffusion in implicit networks. In *2010 IEEE International Conference on Data Mining*, pages 599–608. IEEE.
- [Yin et al., 2012] Yin, P., Luo, P., Wang, M., and Lee, W.-C. (2012). A straw shows which way the wind blows: ranking potentially popular items from early votes. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 623–632. ACM.